CONTROLTAC: Force- and Position-Controlled Tactile Data Augmentation with a Single Reference Image

Dongyu Luo*† Kelin Yu* Amir-Hossein Shahidzadeh
Cornelia Fermuller Yiannis Aloimonos Ruohan Gao

University of Maryland, College Park
lewisluo@conncet.hku.hk kyu85@umd.edu amirsh@umd.edu
fer@cfar.umd.edu yiannis@cs.umd.edu rhgao@umd.edu

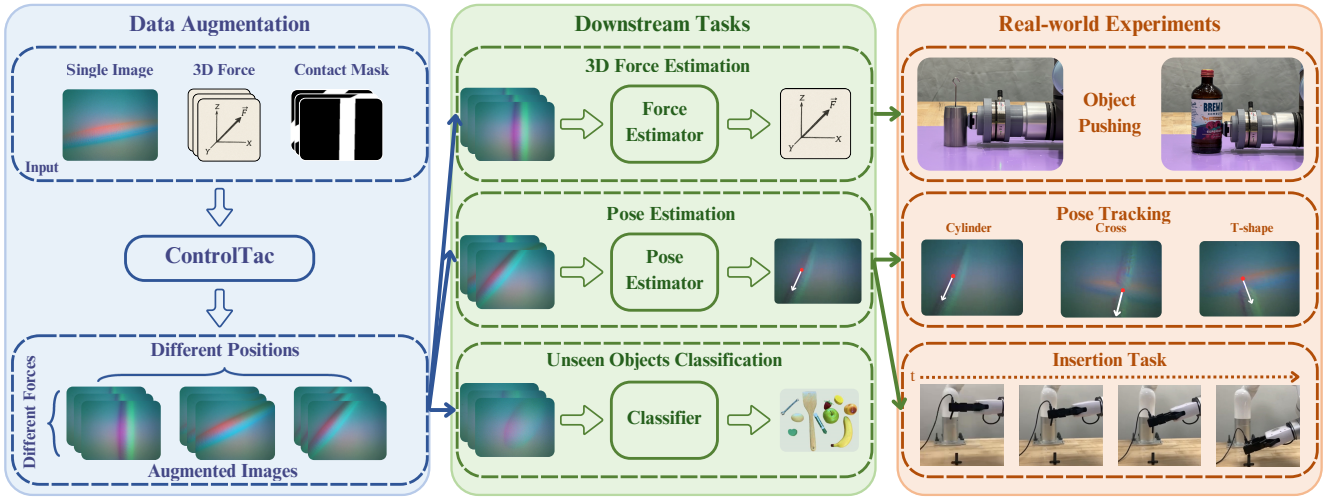

Figure 1. **Illustrations of CONTROLTAC's utilities:** starting from a single reference image, CONTROLTAC can generate tens of thousands of augmented tactile images with various contact forces and contact positions (Left). These augmented images can then be used for various downstream tasks (Middle) and deployed in three real-world experiments (Right).

Abstract

Vision-based tactile sensing has been widely used in perception, reconstruction, and robotic manipulation. However, collecting large-scale tactile data remains costly due to the localized nature of sensor-object interactions and inconsistencies across sensor instances. Existing approaches to scaling tactile data, such as simulation and free-form tactile generation, often suffer from unrealistic output and poor transferability to downstream tasks. To address this, we propose CONTROLTAC*, a two-stage controllable framework that generates realistic tactile images conditioned on a single reference tactile image, contact force, and contact position. With those physical priors as control input,* CONTROLTAC *generates physically plausible and varied tactile images that can be used for effective data augmentation. Through experiments on three downstream tasks, we demonstrate that* CONTROLTAC *can effectively augment tactile datasets and lead to consistent gains. Our three real-world experiments further validate the practical utility of our approach.*

1. Introduction

Vision-based tactile sensing is widely used in material classification [19, 35], 3D reconstruction [28, 58, 59], and robotic manipulation [12, 33, 71]. However, collecting tactile data is costly since it requires physical contact, and the resulting images often vary due to sensor differences, gel instability, and lighting, making them noisy and hard to reuse. This

*These authors contributed equally to this work.
†Dongyu is affiliated with The University of Hong Kong. The work was done during an internship at the University of Maryland.

	Realism	Variation	Controllable
Text2Tac [62, 70]	Low	Low	✗
Vis2Tac [14, 36, 76]	Low	Medium	✗
Simulation [52, 53, 63]	Medium	Medium	✓
CONTROLTAC	High	High	✓

Figure 2. Comparison of tactile data generation approaches. We evaluate realism, output variation, and controllability.

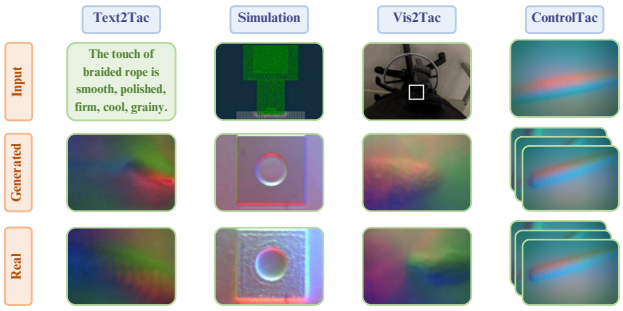

Figure 3. Illustration of CONTROLTAC 's two-stage tactile generation framework.

motivates the need for efficient tactile data augmentation.

Traditional augmentations like color jittering, translation, and rotation have limited effect due to the high variability in tactile images [41, 66]. To scale datasets, two approaches are common: *simulation-based* and *generative* methods. Simulation-based methods [52, 53, 63] model sensor-object interactions but often produce unrealistic images due to imperfect physics. Generative methods [14, 62, 68, 76] synthesize tactile images from text or visual cues but typically lack physical constraints, resulting in low-fidelity outputs. These methods are mainly useful for pre-training [21, 24, 75] or simple tasks like contact localization [14, 19].

We argue that realistic tactile generation requires structured constraints and physical priors. Inspired by ControlNet [74], which improves visual generation via edge and depth conditioning, we propose conditioning tactile generation on contact-relevant factors—force, location, and shape—along with a single reference image. This provides structural cues at minimal data cost and enables physically plausible tactile generation.

We introduce CONTROLTAC, a two-stage tactile generation framework. Stage 1 uses a reference image and 3D force vector to generate a target image with realistic deformation and texture. Stage 2 employs a ControlNet-style module with a 2D contact mask for precise positional refinement. This allows generating diverse, physically plausible tactile images from a single reference, while modeling priors with minimal data. As shown in Fig. 3, CONTROLTAC achieves high realism, diversity, and controllability.

Experiments show that CONTROLTAC improves performance in force estimation, pose estimation, and object classification, often surpassing models trained on much larger real datasets. It generalizes to unseen objects and performs well in real-world tasks, including precise insertion.

Our contributions are: `(1)` A two-stage controllable tactile generation framework for realistic data augmentation; `(2)` Significant gains on multiple downstream tasks using only a single reference image; `(3)` Deployment in real-world robotic experiments with strong performance on precise object insertion.

More related work can be found in Appendix A.

2. Methodology

We propose CONTROLTAC, a controllable tactile image generation framework that generates realistic tactile images from a single reference image, conditioned on both contact force and contact position. This enables scalable tactile data augmentation for a variety of downstream tasks, including force estimation, contact pose estimation, and object classification. By combining physical priors with visual cues, CONTROLTAC ensures that generated images are visually realistic and physically consistent. The full technical details, including network architecture, training procedure, and contact mask definition, are provided in Appendix B.

The core of CONTROLTAC is a two-stage conditional generation pipeline (Fig. 4):

1. **Force-Control Generation:** A conditional diffusion model synthesizes tactile images that reflect a desired target force while preserving the texture and color of a reference image. The reference image is first encoded into a latent representation, and the model is conditioned on a relative 3D force vector $\mathbf{\Delta F}$, defined as the difference between the target and initial forces. Using a diffusion transformer backbone with DDIM sampling, the generator learns realistic force-induced deformations. Training is performed on datasets with ground-truth 3D force annotations to ensure physical consistency. Appendix B.1 provides additional information on force-control generation.
2. **Position-Control Generation:** To control contact position, the pretrained force-control generator is fine-tuned using *contact masks* via ControlNet. Each mask is a fixed, per-object template derived from a lightly pressed reference image, representing the object footprint. Changes in position are applied through rigid in-plane transformations (translation + rotation). During both training and inference, the mask is treated as a latent position-control signal, guiding the generator while preserving force-induced visual patterns. This allows flexible manipulation of object position and orientation without inter-

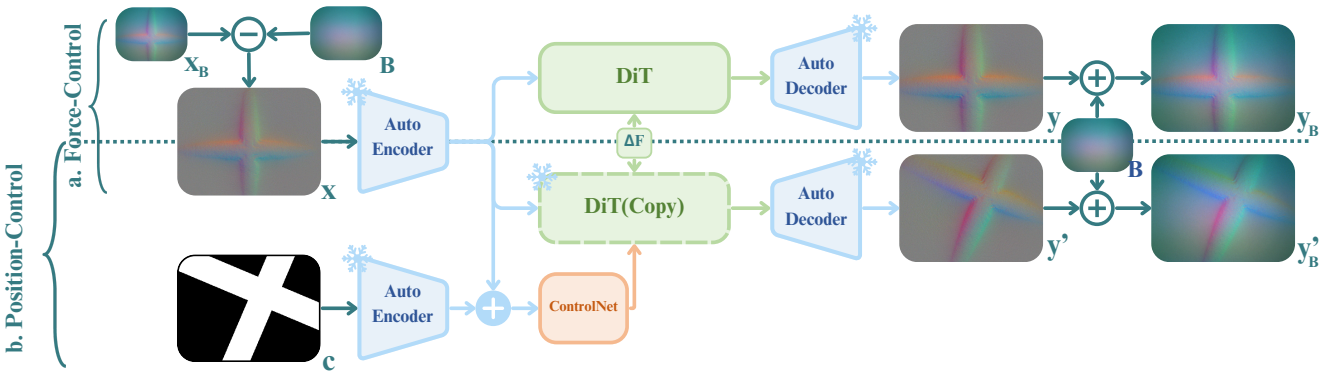

Figure 4. **Overview of our controllable tactile generation framework.** The Force-Control generator synthesizes tactile images conditioned on force, while the Position-Control generator fine-tunes it with ControlNet to control contact position. This two-stage approach enables both realistic and physically consistent tactile image generation.

fering with learned force representations. Further details on position-control generation and the contact mask are provided in Appendix B.1.

This two-stage design offers several advantages. Force and position are controlled independently, improving training stability and generation predictability. In addition, generated tactile images can augment datasets for diverse downstream tasks without requiring additional data collection.

2.1. Data Augmentation for Downstream Tasks

Once trained, CONTROLTAC generates tactile images to support three representative downstream tasks. Additional details can be found in Appendix B.2.

- **Force Estimation:** Predicting 3D force vectors from tactile images. The model uses a pretrained ViT encoder and a regressor-decoder architecture to learn correlations between tactile images and forces.
- **Contact Pose Estimation:** Estimating contact location and orientation. Using the same encoder, the output predicts x, y coordinates and object rotation, with supervision from the contact mask instead of depth.
- **Object Classification:** Recognizing object identity from tactile images. Both CNN and ViT-based classifiers are evaluated, including pretrained ViTs. Augmentation improves classifier performance across six objects: five from the FeelAnyForce dataset [50] and one additional T-shaped object.

By explicitly modeling both force and contact position, CONTROLTAC generates large-scale, physically-consistent tactile datasets that improve performance across a variety of tactile perception tasks. Full architectural details, training procedures, and additional implementation specifics are provided in Appendix B.

3. Experiments

We evaluate our framework through three main aspects: (1) **tactile image generation** (Sec.3.1 and App. E.1) to assess generation quality against baselines; (2) **downstream tasks** including force estimation (Sec.3.2 and App. E.2), pose estimation (Sec.3.3 and App. E.3), and object classification (Sec.3.4 and App. E.4) to verify the effectiveness of generated tactile data for model training; and (3) **real-world experiments** (Sec.3.5 and App. E.5) including object pushing, real-time pose tracking, and precise insertion, demonstrating practical applicability. Full experimental details are provided in Appendix E.

3.1. Generation Quality Evaluation

We compare our two-stage conditional tactile generation framework (CONTROLTAC) with three baselines: hybrid force-position conditional diffusion, separate-control pipeline, and single-stage force-control generation. Evaluation is performed using SSIM and pixel-wise MSE on real tactile test data from FeelAnyForce [50].

Table 1. Comparison in MSE and SSIM. Hybrid represents Hybrid Force-Position Conditional Diffusion Model, and Separate represents Separate-Control Pipeline.

Method	MSE ↓	SSIM ↑
Hybrid	31	0.81
Separate	157	0.79
Ours (First Stage)	**18**	**0.84**
Ours	23	0.83

Our method achieves the best balance of structural and pixel-level quality. The two-stage framework enables accurate control over both force and contact position, avoiding error accumulation observed in separate-control pipelines. Qualitative visualizations in Fig. 6 show realistic gel deformation and brightness patterns. Appendix E.1 provides more detailed information on generation quality evaluation.

3.2. Downstream Task: Force Estimation

Training a force estimator with augmented data significantly reduces MAE compared to using limited real data alone

(Fig.5,7). Augmenting 1,000 real samples with generated images covering 20–40 force levels achieves comparable performance to training on the full real dataset (20,000 samples). Position-control generation further improves performance when angular coverage in the real dataset is limited, demonstrating that CONTROLTAC effectively enriches force distributions across contact positions. See Appendix E.2 for more details on force estimation.

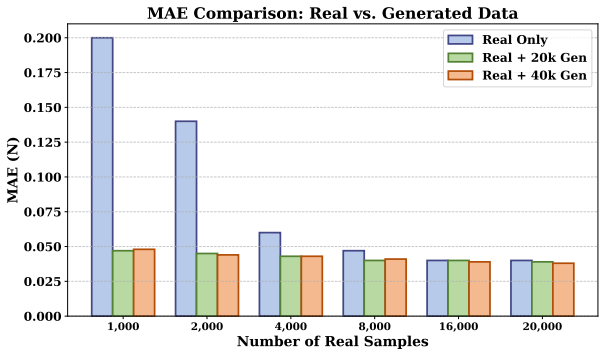

Figure 5. Force estimation performance (MAE) across different quantities of real and generated data. The force range is 1–10 N.

3.3. Downstream Task: Pose Estimation

Pose estimators trained solely on generated images achieve strong performance for both seen and unseen objects, outperforming simulation-based augmentation (Taxim [52]) and PCA-based methods (Table 2). Using varying forces further improves accuracy compared to fixed-force settings. Combining a small amount of real data with generated images often yields the best results. Additional information regarding pose estimation is provided in Appendix E.3.

3.4. Downstream Task: Object Classification

Generated data improves classification accuracy across CNN and ViT models compared to geometric and color-based augmentations (Table 3). Even with small datasets, CONTROLTAC significantly enhances ViT performance, showing that conditional tactile generation provides diverse and realistic tactile representations. Appendix E.4 provides more detailed information on object classification.

3.5. Real-world Experiments

We deploy trained force and pose estimators in object pushing, real-time pose tracking, and insertion. Further details on real-world experiments can be found in Appendix E.1.

Object Pushing: Force estimators trained on generated data achieve comparable performance to those trained on real data, demonstrating effective generalization to real objects (Table 4).

Real-time Pose Tracking: Pose estimators track objects at 10 Hz, validating real-time applicability.

Insertion Task: Generated-data-trained models achieve 90% (cylinder), 85% (cross/T-shape), and 75% (Type-C USB) success rates, demonstrating strong practical utility with 3 mm tolerance.

Table 2. Pose estimation errors (in pixels and degrees) under different settings.

Training Set	X ↓	Y ↓	Angle ↓
Cylinder (3 Types)			
PCA	15	13	22
6,000 real	8	8	4
36,000 sim	18	15	6
6,000 gen (unfixed)	7	6	3
Cross			
PCA	56	19	18
3,000 real	6	6	2
12,000 sim	18	19	5
3,000 gen (unfixed)	4	5	2
T-shape (Unseen)			
1,000 gen (unfixed)	5	5	4
4,000 gen (unfixed)	4	5	2
USB (Unseen)			
1,000 gen (unfixed)	12	11	4
4,000 gen (unfixed)	8	9	3

Table 3. Accuracy comparisons across models and augmentation methods. G: geometric data augmentation; C: color augmentation; Gen: our CONTROLTAC-based augmentation method.

	2400 (G)	4800 (G)	2400 (G+C)	4800 (G+C)	2400 (Gen)	4800 (Gen)
CNN	0.74	0.68	0.65	0.69	0.85	**0.87**
ViT (Scratch)	0.62	0.60	0.62	0.65	0.93	**0.95**
ViT (ImageNet)	0.78	0.76	0.74	0.79	0.99	**0.99**

Table 4. Results of object pushing experiments for the four objects.

Force [N]	Weight (1.0)	Cyl. (0.50)	Cyl. (0.56)	Bottle (0.63)
Force ATI (G.T.)	2.24	0.96	1.06	1.08
Force (Real Data)	2.38	1.08	1.18	1.14
Force (Ours)	2.36	1.11	1.17	1.16

4. Conclusion

We present CONTROLTAC, a two-stage conditional tactile generation framework that produces realistic and diverse tactile images from a single reference, conditioned on force and contact position. Experiments on three downstream tasks and in real-world settings show that CONTROLTAC effectively supports data augmentation and performs well in practical applications. While this work is the first attempt at controllable tactile image generation, it currently considers only force and position, omitting other physical parameters like surface texture and material hardness. Future work will extend the framework to incorporate additional conditions, leveraging its modular design for richer, more physically grounded tactile generation.

References

[1] Arpit Agarwal, Tim Man, and Wenzhen Yuan. Simulation of vision-based tactile sensors using physics based rendering, 2021. 8

[2] Bo Ai, Stephen Tian, Haochen Shi, Yixuan Wang, Cheston Tan, Yunzhu Li, and Jiajun Wu. Robopack: Learning tactile-informed dynamics models for dense packing, 2024. 8

[3] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pages 2745–2754, 2017. 9

[4] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. Reskin:versatile, replaceable, lasting tactile skins. In *CoRL*, 2021. 8

[5] Raunaq Bhirangi, Venkatesh Pattabiraman, Enes Erciyes, Yifeng Cao, Tess Hellebrekers, and Lerrel Pinto. Anyskin: Plug-and-play skin sensing for robotic touch, 2024. 8

[6] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H. Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018. 8

[7] Roberto Calandra, Andrew Owens, Manu Upadhyaya, Wenzhen Yuan, Justin Lin, Edward H. Adelson, and Sergey Levine. The feeling of success: Does touch sensing help predict grasp outcomes?, 2025. 8

[8] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 9

[9] Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. Pixart-{\delta}: Fast and controllable image generation with latent consistency models. *arXiv preprint arXiv:2401.05252*, 2024. 9

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 10, 17

[11] Siyuan Dong and Alberto Rodriguez. Tactile-based insertion for dense box-packing, 2019. 8

[12] Siyuan Dong, Devesh K. Jha, Diego Romeres, Sangwoon Kim, Daniel Nikovski, and Alberto Rodriguez. Tactile-rl for insertion: Generalization to objects of unknown geometry, 2021. 1, 8

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 10, 17

[14] Yiming Dou, Fengyu Yang, Yi Liu, Antonio Loquercio, and Andrew Owens. Tactile-augmented radiance fields. *arXiv preprint arXiv:2405.04534*, 2024. 2, 8

[15] Ruoxuan Feng, Jiangyu Hu, Wenke Xia, TianciGao, Ao Shen, Yuhao Sun, Bin Fang, and Di Hu. Anytouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors. In *The Thirteenth International Conference on Learning Representations*, 2025. 8

[16] Ethan Fetaya, Jörn-Henrik Jacobsen, Will Grathwohl, and Richard Zemel. Understanding the limitations of conditional generative models. *arXiv preprint arXiv:1906.01171*, 2019. 9

[17] Ruohan Gao, Yen-Yu Chang, Shivani Mall, Li Fei-Fei, and Jiajun Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. In *CoRL*, 2021. 8

[18] Ruohan Gao*, Zilin Si*, Yen-Yu Chang*, Samuel Clarke, Jeannette Bohg, Li Fei-Fei, Wenzhen Yuan, and Jiajun Wu. Objectfolder 2.0: A multisensory object dataset for sim2real transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

[19] Ruohan Gao, Yiming Dou, Hao Li, Tanmay Agarwal, Jeannette Bohg, Yunzhu Li, Li Fei-Fei, and Jiajun Wu. The objectfolder benchmark: Multisensory learning with neural and real objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17276–17286, 2023. 1, 2, 8

[20] Ruihan Gao, Kangle Deng, Gengshan Yang, Wenzhen Yuan, and Jun-Yan Zhu. Tactile dreamfusion: Exploiting tactile sensing for 3d generation, 2024. 8

[21] Harsh Gupta, Yuchen Mo, Shengmiao Jin, and Wenzhen Yuan. Sensor-invariant tactile representation. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 8

[22] Yunhai Han, Kelin Yu, Rahul Batra, Nathan Boyd, Chaitanya Mehta, Tuo Zhao, Yu She, Seth Hutchinson, and Ye Zhao. Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer. *IEEE/ASME Transactions on Mechatronics*, 30(1):554–566, 2025. 8

[23] Carolina Higuera, Byron Boots, and Mustafa Mukadam. Learning to read braille: Bridging the tactile reality gap with diffusion models, 2023. 8

[24] Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In *8th Annual Conference on Robot Learning*, 2024. 2, 8

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 9

[26] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3d vitac:learning fine-grained manipulation with visuo-tactile sensing. In *Proceedings of Robotics: Conference on Robot Learning(CoRL)*, 2024. 8

[27] Hung-Jui Huang, Xiaofeng Guo, and Wenzhen Yuan. Understanding dynamic tactile sensing for liquid property estimation, 2022. 8

[28] Hung-Jui Huang, Michael Kaess, and Wenzhen Yuan. Normalflow: Fast, robust, and accurate contact-based object 6dof pose tracking with vision-based tactile sensors. *IEEE Robotics and Automation Letters*, pages 1–8, 2024. 1, 8

[29] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 9

[30] Abdul Jabbar, Xi Li, and Bourahla Omar. A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys (CSUR)*, 54(8):1–49, 2021. 9

[31] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 9

[32] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. 8

[33] Hao Li, Yizhi Zhang, Junzhe Zhu, Shaoxiong Wang, Michelle A. Lee, Huazhe Xu, Edward Adelson, Li Fei-Fei, Ruohan Gao, and Jiajun Wu. See, hear, and feel: Smart sensory fusion for robotic manipulation. In *CoRL*, 2022. 1, 8

[34] Jianhua Li, Siyuan Dong, and Edward Adelson. Slip detection with combined tactile and visual information. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7772–7777, 2018. 8

[35] Rui Li and Edward H. Adelson. Sensing and recognizing surface textures using a gelsight sensor. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1241–1247, 2013. 1, 8

[36] Yunzhu Li, Jun-Yan Zhu, Russ Tedrake, and Antonio Torralba. Connecting touch and vision via cross-modal prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 8

[37] Changyi Lin, Han Zhang, Jikai Xu, Lei Wu, and Huazhe Xu. 9dtact: A compact vision-based tactile sensor for accurate 3d shape reconstruction and generalizable 6d force estimation. *arXiv preprint arXiv:2308.14277*, 2023. 8

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 10

[39] Dongyu Luo, Jianyu Wu, Jing Wang, Hairun Xie, Xiangyu Yue, and Shixiang Tang. Diffffluid: Plain diffusion models are effective predictors of flow dynamics. *arXiv preprint arXiv:2409.13665*, 2024. 9

[40] Daolin Ma, Elliott Donlon, Siyuan Dong, and Alberto Rodriguez. Dense tactile force distribution estimation using gelslim and inverse fem, 2019. 8

[41] Philip Maus, Jaeseok Kim, Olivia Nocentini, Muhammad Zain Bashir, and Filippo Cavallo. The impact of data augmentation on tactile-based object classification using deep learning approach. *IEEE Sensors Journal*, 22(14):14574–14583, 2022. 2, 13

[42] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 9

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 10

[44] Achraf Oussidi and Azeddine Elhassouny. Deep generative models: Survey. In *2018 International conference on intelligent systems and computer vision (ISCV)*, pages 1–8. IEEE, 2018. 9

[45] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 9

[46] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *7th Annual Conference on Robot Learning*, 2023. 8

[47] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016. 9

[48] Samanta Rodriguez, Yiming Dou, Miquel Oller, Andrew Owens, and Nima Fazeli. Touch2touch: Cross-modal tactile generation for object manipulation, 2024. 8

[49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 9

[50] Amir-Hossein Shahidzadeh, Gabriele Caddeo, Koushik Alapati, Lorenzo Natale, Cornelia Fermuller, and Yiannis Aloimonos. Feelanyforce: Estimating contact force feedback from tactile sensation for vision-based tactile sensors, 2024. 3, 8, 9, 10, 11, 12, 15, 16

[51] Amir-Hossein Shahidzadeh, Seong Jong Yoo, Pavan Mantripragada, Chahat Deep Singh, Cornelia Fermüller, and Yiannis Aloimonos. Actexplore: Active tactile exploration on unknown objects. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3411–3418, 2024. 8

[52] Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):2361–2368, 2022. 2, 4, 8, 12, 13, 18, 21

[53] Zilin Si, Gu Zhang, Qingwei Ben, Branden Romero, Zhou Xian, Chao Liu, and Chuang Gan. DIFFTACTILE: A physics-based differentiable tactile simulator for contact-rich robotic manipulation. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 8

[54] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015. 9

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 9

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 9

[57] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569(7758), 2019. 8

[58] Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, Joseph Ortiz, and Mustafa

Mukadam. Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation. *Science Robotics*, page adl0628, 2024. 1, 8

[59] Aiden Swann, Matthew Strong, Won Kyung Do, Gadiel Sznaier Camps, Mac Schwager, and Monroe Kennedy. Touchgs: Visual-tactile supervised 3d gaussian splatting. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10511–10518, 2024. 1, 8

[60] Ian Taylor, Siyuan Dong, and Alberto Rodriguez. Gelslim3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger, 2021. 8

[61] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017. 9

[62] Jiahang Tu, Hao Fu, Fengyu Yang, Hanbin Zhao, Chao Zhang, and Hui Qian. Texttoucher: Fine-grained text-to-touch generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7455–7463, 2025. 2, 8

[63] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):3930–3937, 2022. 2, 8

[64] Ross Wightman. Pytorch image models. `https://github.com/huggingface/pytorch-image-models`, 2019. 17

[65] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 9, 10

[66] Gang Yan, Jun Yuyeol, Satoshi Funabashi, Tito Pradhono Tomo, Sophon Somlor, Alexander Schmitz, and Shigeki Sugano. Geometric transformation: Tactile data augmentation for robotic learning. In *2023 IEEE International Conference on Development and Learning (ICDL)*, pages 346–353. IEEE, 2023. 2, 13

[67] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 776–791. Springer, 2016. 9

[68] Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan, and Andrew Owens. Touch and go: Learning from human-collected vision and touch. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 2, 8

[69] Fengyu Yang, Jiacheng Zhang, and Andrew Owens. Generating visual scenes from touch. *International Conference on Computer Vision (ICCV)*, 2023. 8

[70] Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, and Alex Wong. Binding touch to everything: Learning unified multimodal tactile representations, 2024. 2, 8

[71] Kelin Yu, Yunhai Han, Qixian Wang, Vaibhav Saxena, Danfei Xu, and Ye Zhao. Mimictouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. In *8th Annual Conference on Robot Learning*, 2024. 1, 8

[72] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017. 8

[73] Wenzhen Yuan, Chenzhuo Zhu, Andrew Owens, Mandayam A. Srinivasan, and Edward H. Adelson. Shape-independent hardness estimation using deep learning and a gelsight tactile sensor. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, page 951–958. IEEE, 2017. 8

[74] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2, 9, 11

[75] Jialiang Zhao, Yuxiang Ma, Lirui Wang, and Edward H. Adelson. Transferable tactile transformers for representation learning across diverse sensors and tasks, 2024. 2, 8

[76] Shaohong Zhong, Alessandro Albini, Oiwi Parker Jones, Perla Maiolino, and Ingmar Posner. Touching a neRF: Leveraging neural radiance fields for tactile sensory data generation. In *6th Annual Conference on Robot Learning*, 2022. 2, 8

[77] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 9

Appendix Overview

In this appendix, we provide additional details and analyses to complement the main paper. The appendix is organized as follows:

- **Related Work (Sec. A):** We provide a more detailed discussion on vision-based tactile sensing, tactile dataset collection, tactile image generation, and conditional image generation. This section situates our work in the context of prior literature and highlights the challenges our method addresses.
- **Detailed Methodology (Sec. B):** We present the complete architecture and design of CONTROLTAC, including the two-stage conditional tactile generation framework, force-control and position-control generators, and the use of contact masks. We also describe how generated data can be leveraged for downstream tasks such as force estimation, contact pose estimation, and object classification.
- **Implementation Details (Sec. C):** Training configurations, hyperparameters, and evaluation metrics are detailed to facilitate reproducibility.
- **Details of Baselines (Sec. D):** We describe the baseline methods used for comparison, including the hybrid force-position conditional diffusion model and the separate-control model.
- **Details of Experiments (Sec. E):** This section includes extensive experimental evaluations, covering generation quality, downstream task performance, and real-world deployment. We provide quantitative results, qualitative visualizations, and comparisons with baseline methods.
- **Additional Experiments (Sec. F):** This section contains robustness analyses, data composition studies, and the effect of varying contact position counts on model performance.
- **Classifier Architectures (Sec. G):** Details of CNN and Vision Transformer (ViT) classifiers used for object classification are provided, including layer configurations and pretraining information.
- **Details of Precise Insertion (Sec. H):** Specifications and setup for the robotic insertion tasks, including both object-based and Type-C USB insertions, are described.
- **Failure Analysis (Sec. I):** We discuss limitations of our model, particularly for objects with flat surfaces or rich textures, and illustrate how small augmentations can improve performance.
- **Additional Visualizations (Sec. J):** We provide supplementary figures illustrating error maps, generated tactile images under force control, simulated tactile images, and object classification examples.

This appendix serves to provide a comprehensive understanding of our methodology, experimental setup, and additional analyses supporting the claims made in the main paper.

A. Related Work

Vision-based Tactile Sensing. Recently, various tactile sensors have been used in different scenarios, such as vision-based tactile sensors [32, 37, 60, 72], magnetic tactile sensors [4, 5], and piezo-resistive tactile sensors [26, 57]. In this paper, we focus mainly on vision-based tactile sensor, which has the highest resolution and can be used to detect precise textures [19, 35] and shear forces [40, 50].

Because of its high resolution, vision-based tactile sensors have been widely utilized in different perception tasks, such as liquid property classification [27], hardness classification [73], 3D reconstruction [28, 58, 59], 3D generation [20, 51], slip detection [34], and pose estimation [28]. Also, it has been used for various robotic tasks, such as grasping [6, 7, 22], insertion [12, 33, 71], pouring [33], in-hand rotation [46], and dense packing [2, 11, 71]. However, the lack of data remains a major challenge for vision-based tactile sensing because collecting local contact on diverse objects is expensive. In this paper, we introduce a new framework for scaling up tactile datasets in downstream tasks with conditional tactile generation.

Building Tactile Datasets. To address the scarcity of tactile data, many prior works focus on collecting large-scale real-world datasets [14, 17, 36, 68]. While these efforts help scale up tactile data, the quantity remains limited, and the resulting datasets are often difficult to reuse fordownstream tasks—especially in robotics tasks—due to significant variability across sensors.

Another approach is to use simulation [1, 18, 52, 53, 63], which have been widely adopted for pre-training [15, 21, 24, 75] and Sim2Real transfer [18, 23, 53]. However, bridging the Sim2Real gap remains a major challenge, as illustrated in Fig. 6. High-quality Sim2Real transfer typically still requires large real datasets for co-training [63] or the use of generative models for domain adaptation [23]. To this end, we propose a controllable tactile generation model that can scale existing tactile datasets under different physical conditions.

Tactile Image Generation. Text-to-tactile generation [62, 70] and vision-to-tactile generation [14, 36, 68, 69, 76] have been widely used for representation learning [15, 75], contact localization [14, 19], classification [14, 76], and retrieval [19, 70]. Cross-sensor generation [48] has also been explored to utilize various properties of different tactile sensors. However, as shown in Fig. 3, the free-form generation from visual images often yields low-quality outputs, limiting its utility in more complex downstream tasks. To address this, we propose a conditional diffusion model that generates tactile images for data augmentation, guided by physical constraints and priors. We present both analysis and qualitative examples in Fig. 3 to highlight the limitations of existing approaches and the strengths of our method.

Conditional Image Generation. Conditional image

generation has become a central topic in generative modeling, where the goal is to generate images guided by structured inputs such as class labels, text, or physical parameters. Early methods [3, 29, 47, 61, 67, 77] based on conditional GANs [42] and conditional VAEs [54] demonstrate the feasibility of conditioning image generation on external inputs but often suffer from limitations in image quality and training stability [16, 30, 31, 44]. More recently, diffusion models [9, 25, 39, 49, 55, 56, 65] have emerged as state-of-the-art approaches due to their ability to generate high-fidelity and diverse images through a gradual denoising process. Meanwhile, ControlNet [74] enhances diffusion-based models by incorporating an auxiliary network that injects explicit structural conditions—such as edge maps, depth maps, or human poses—into the generation pipeline. This allows for fine-grained control over the output while maintaining the quality and diversity of diffusion models. Inspired by, but distinct from the prior work above, we tackle the new problem of controllable tactile image generation.

B. Detailed Methodology

We present CONTROLTAC, a controllable framework for generating realistic tactile images to scale up tactile dataset in downstream tasks, using only a single reference tactile image along with contact force and contact position. The key innovation lies in leveraging the reference tactile image to preserve contact texture and color, while incorporating physical conditions—force and contact location—through a two-stage conditional tactile generation pipeline. This design ensures the generated tactile images are both realistic and physically consistent, enabling effective tactile data augmentation.

In this section, we first introduce the architecture of our two-stage conditional tactile generation framework, which includes a force-control generator and a position-control generator (Sec. B.1). Then, we demonstrate how to effectively leverage the generated data for downstream tasks such as force estimation, contact pose estimation, and object classification (Sec. B.2). The overall architecture of our framework is illustrated in Fig. 4.

B.1. Two-stage Conditional Tactile Generation Framework

We propose a two-stage conditional tactile image generation framework that incorporates force and contact position as controllable physical priors. The model also leverages a reference tactile image to preserve color and texture cues. `(1)` In the first stage, the force-control generator takes a reference tactile image and the relative force as input to generate a target image that reflects the desired force. `(2)` In the second stage, we fine-tune the pretrained force-control generator with contact masks using ControlNet [74] to control the contact position of generated tactile images.

Force-Control Generation. To generate a tactile image corresponding to a target force, we train a conditional diffusion model defined as $\mathbf{y} = \mathcal{D}(\mathcal{F}_f(\mathbf{z}^{(\mathbf{x})}), \mathbf{\Delta F})$, where the Diffusion Transformer (DiT) [45] is used as the backbone and DDIM [55] serves as the sampler to improve both generation quality and inference efficiency. In this formulation, the force-control generator is denoted by $\mathcal{F}_f(\cdot)$. The target tactile image is represented by $\mathbf{y} \in \mathbb{R}^{W\times H\times 3}$, while the reference tactile image is denoted as $\mathbf{x} \in \mathbb{R}^{W\times H\times 3}$. The latent representation $\mathbf{z}^{(\mathbf{x})} = \mathcal{E}(\mathbf{x})$ is obtained by encoding $\mathbf{x}$ using the frozen pretrained encoder $\mathcal{E}(\cdot)$ introduced in SANA [65]. The output of the diffusion model is subsequently decoded using the corresponding frozen pretrained decoder $\mathcal{D}(\cdot)$, also proposed in SANA [65]. The model is conditioned on a relative force vector $\mathbf{\Delta F} \in \mathbb{R}^3$, computed as the difference between the desired target force $\mathbf{F}_t$ and the initial force $\mathbf{F}_i$ associated with the reference image: $\mathbf{\Delta F} = \mathbf{F}_t - \mathbf{F}_i$. To enable force-guided generation, we train the force-control generator $\mathcal{F}_f(\cdot)$ using the dataset proposed in FeelAnyForce [50], which provides ground-truth annotations of 3D force vectors.

Position-Control Mask. We represent the position-control signal with a compact, per-object binary template that we call the contact mask. Notably, This mask is a global, object-level template but not a per-frame local contact patch. It is computed *once* from a lightly pressed reference tactile image (initial contact) and then kept fixed for that object (see Fig. 6). Consequently, its shape is independent of the applied force. During training and inference, we never re-estimate the mask from the current frame. Changes in contact *position* are encoded solely by applying a rigid in-plane transform (translation + rotation) to this fixed template. For accuracy, we manually register each mask to its reference tactile image with ± 1 pixel translational and $\pm 1^\circ$ rotational precision. Additional robustness studies are provided in Appendix F.1. This representation avoids common pitfalls when defining contact position: `(1)` it avoids the ambiguity of using a center point (x, y) plus an angle, which becomes ill-posed when the object footprint exceeds the sensor area; `(2)` it avoids the inconsistency of depth map or edge-based methods (e.g., Canny [8]), whose detected boundaries can vary with the pressed region and force. In summary, "mask" denotes a fixed per-object template for position control, while force-induced local area changes are <u>not</u> encoded by the mask.

Position-Control Generation. Building upon the pretrained force-control generator $\mathcal{F}_f(\cdot)$, we utilize the ControlNet [74] to fine-tune the force-control generator using the contact mask as a control signal. Specifically, we follow the approach from PixArt-δ [9], where the ControlNet is applied to the first half of the DiT [45] blocks. The output of each block is added to the output of the corresponding frozen block, serving as the input to the next frozen block. The generated tactile image $\mathbf{y}'$ satisfying the target

force and target contact position is obtained from the model $\mathbf{y}' = \mathcal{D}(\mathcal{F}_c(\mathbf{z}^{(\mathbf{x})}, \mathbf{z}^{(\mathbf{c})}, \mathbf{\Delta F}))$, where $\mathcal{F}_c(\cdot)$ is the generator built upon the force-control generator with the ControlNet. We treat the contact mask $\mathbf{c} \in \mathbb{R}^{W \times H \times 1}$ as the position-control signal, and $\mathbf{z}^{(\mathbf{c})} = \mathcal{E}(\mathbf{c})$ is the latent representation of $\mathbf{c}$ obtained from the autoencoder of SANA [65]. We train the model on the aligned contact masks from FeelAnyForce [50].

B.2. Data Augmentation for Downstream Tasks

After training our two-stage conditional tactile generation framework, we apply the generated tactile images for data augmentation. Broadly, the generated images support data augmentation in three settings: tasks with force labels, tasks with pose labels, and tasks where labels remain unchanged after augmentation. Specifically, we select three tasks that require both realistic and large-scale data for effective augmentation: force estimation, contact pose estimation, and object classification.

Force Estimation. In this task, we adopt the force estimation framework proposed in FeelAnyForce [50], which is based on a ViT [13] encoder pretrained on DINOv2 [43]. The framework takes tactile images as input, consisting of a regressor that predicts the 3D force vector and a decoder that reconstructs the depth image during training to enhance the learning of the tactile-force relationship.

Contact Pose Estimation. In this task, we retain the ViT [13] encoder pretrained on DINOv2 [43] from the force estimator in FeelAnyForce [50]. The regressor output is changed from the 3D force vector to the x and y coordinates of the contact center, as well as the angle of the contact object relative to the tactile sensor. Additionally, the decoder's supervision is shifted from depth to the contact mask. Through these modifications, we obtain a pose estimator.

Object Classification. To ensure fair comparison, we use three common classifiers: a plain CNN, a ViT [13] without pretraining, and a ViT pretrained on ImageNet [10]. The classification task involves six objects: five from the FeelAnyForce dataset [50]—banana, marker, nectarine, ring, and thick cylinder—and one additional object we collected, the T-shape. Appendix J.5 shows the objects and tactile images (Fig. 14), and Appendix G provides more details on the classifiers.

C. Implementation Details

C.1. Training Configuration

Both force-control generation component and position-control generation component are trained using the AdamW [38] optimizer and a cosine annealing learning rate scheduler. For the force-control generator, the learning rate is annealed from an initial value of 1×10^{-4} to a final value of 1×10^{-5}. For the position-control generator, the learning rate decays from 1×10^{-5} to 1×10^{-6}. Each model is trained for 75,000 steps on a single NVIDIA RTX A5000 GPU with a batch size of 4. The loss function used for training is a weighted combination of L1 loss and mean squared error (MSE): $0.5 \times \mathcal{L}_{\text{L1}} + 0.5 \times \mathcal{L}_{\text{MSE}}$.

C.2. Metrics

We evaluate our models using several commonly used metrics, including mean squared error (MSE), L1 loss, mean absolute error (MAE), and structural similarity index measure (SSIM). Specifically, the following metrics are reported:

- **Mean Squared Error (MSE):** $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- **L1 Loss:** $\text{L1} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- **Mean Absolute Error (MAE):** $\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- **Structural Similarity Index Measure (SSIM):** $\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$

D. Details of Baselines

D.1. Hybrid Force-Position Conditional Diffusion Model:

In this approach, we train a diffusion model $\mathbf{y} = \mathcal{D}(\mathcal{F}_h(\mathbf{z}^{(\mathbf{x})}, \mathbf{z}^{(\mathbf{c})}, \mathbf{\Delta f}))$. Here, the latent representation of initial tactile image $\mathbf{z}^{(\mathbf{x})}$, contact mask $\mathbf{z}^{(\mathbf{c})}$, and target force change $\mathbf{\Delta f}$ are simultaneously input into the diffusion model $\mathcal{F}_h(\cdot)$, which is then passed into the autodecoder $\mathcal{D}(\cdot)$ to generate the output $\mathbf{y}$.

D.2. Separate Force-Position Conditional Diffusion Model:

In the first stage, we follow the previous force-control generator method by inputting the latent representation of the initial tactile image $\mathbf{z}^{(\mathbf{x})}$ and the target force change $\mathbf{\Delta f}$ into the force-control generator $\mathcal{F}_f(\cdot)$ to produce a latent representation of the tactile image $\mathbf{z}^{(\mathbf{x}')} = \mathcal{F}_f(\mathbf{z}^{(\mathbf{x})}, \mathbf{\Delta f})$ that satisfies the target force. In the second stage, this generated latent representation $\mathbf{z}^{(\mathbf{x}')}$, along with the latent representation of the contact mask $\mathbf{z}^{(\mathbf{c})}$, is input into the position-control generator $\mathcal{F}_p(\cdot)$. The output $\mathbf{z}^{(\mathbf{y})} = \mathcal{F}_p(\mathbf{z}^{(\mathbf{x}')}, \mathbf{z}^{(\mathbf{c})})$, which satisfies both the target force and contact position, is then decoded to produce the final tactile image $\mathbf{y} = \mathcal{D}(\mathbf{z}^{(\mathbf{y})})$.

E. Details of Experiments

In this section, we evaluate our proposed framework through extensive experiments in tactile image generation, data augmentation in three downstream tasks, and three real-world experiments. We firstly evaluate the generation quality of our two-stage conditional tactile generation framework with two baselines in Sec. E.1. Then, we perform three downstream tasks to evaluate the data augmentation capability of our framework (Sec. E.2, E.3, E.4).

Finally, we deploy the trained force estimator and pose estimator into three real-world experiments (Sec. E.5).

We train the force-control generator component using 20,000 tactile images with corresponding 3D force vectors from FeelAnyForce [50]. The ControlNet for position-control generation is trained using 7,000 tactile images, where each object contributes 300 unique contact positions, also paired with 3D force vectors from FeelAnyForce [50]. More training details are shown in Appendix C.

E.1. Generation Quality Evaluation

In this section, we compare the generation quality of our two-stage conditional tactile generation framework with three baselines: `(1)` a hybrid force-position conditional diffusion model, which trains conditional diffusion with both force and position at the same time; `(2)` a separate-control pipeline, which trains the position-control using the generated images from the pretrained force-control (in the first stage) model as input; `(3)` the first stage of CONTROLTAC, which performs augmentation conditioned only on force. For training data, 7,000 samples are used to train the hybrid force-position conditional diffusion model. For the separate-control pipeline, we use 20,000 samples to train the force-control generator and 7,000 samples to train the position-control generator. Details of the baselines and training procedures for the three models can be found in Appendix D.

Evaluation. To compare the quality of tactile images generated by CONTROLTAC with two other baseline methods and the first stage force-control generation, we calculate SSIM (Structural Similarity) and pixel-wise MSE (Mean Squared Error) on the test data of real tactile images. The test data, which comes from FeelAnyForce [50], includes the same objects as the training set but with different contact positions and forces. In Table 1, we present a comparison of SSIM and MSE for four methods; in Fig. 6, we show tactile images generated by these methods.

Key Findings. From Table 1, we can see that our method outperforms all baselines in both MSE and SSIM. For SSIM, it indicates that CONTROLTAC performs slightly better than the other baselines in terms of structure, gel deformation, and brightness information. For the MSE, which highlights the pixel-level difference and the generation precision, CONTROLTAC clearly outperforms the other baselines. Compared to our single-stage force-control model, our two-stage conditional tactile generation framework achieves comparable performance while enabling additinoal control over contact posision. For the hybrid framework, the lower performance is mainly due to that force control requires more data than position control. As a result, using ControlNet [74] to finetune the pre-trained force control generator on a small amount of position data yields better performance. In the separate-control pipeline, errors from both the force-control and position-control generators accumulate, resulting in significantly worse overall performance. Additionally, we provide a detailed failure analysis in Appendix I.

E.2. Downstream Task: Force Estimation

In this section, to demonstrate that CONTROLTAC can generate realistic tactile images corresponding to the target force, we validate it by training a force estimator using the generated tactile images.

We first evaluate the effectiveness of the force-control generator in CONTROLTAC by comparing the performance of force estimators trained on various combinations of real and generated data. We use 1,000 different contact positions from the dataset as the real dataset and augment 20 or 40 forces with the force-control generator, resulting in datasets of 20,000 and 40,000 generated samples, respectively. We then evaluate the performance by co-training on varying amounts of real data (from 1,000 real data to 20,000 real samples) combined with the augmented data.

As shown in Fig. 5, augmenting the real data (1,000 samples) with a larger amount of generated data significantly reduces the MAE compared to using the real data alone. Moreover, with the generated dataset, the model achieves the same performance as training on the full real dataset (20,000 images) with only 8,000 real images. This suggests that the generated data effectively enrich the force distribution at each contact position, enhancing the training of the force estimator. Furthermore, combining larger quantities of both real and generated data yields the best performance, which highlights the realism and utility of the generated data.

After validating the force-control generator in CONTROLTAC, we evaluate the full framework. We use CONTROLTAC to generate 15,000 or 30,000 tactile images with 750 different positions. We evaluate the performance by adding those augmented images to different number of real data for co-training (from 1,000 to 15,000 real samples).

To demonstrate that training a high-performance force estimator requires covering different contact positions, we divided the real data according to the angles because the color of tactile images varies across different contact angles. Visualization of different angles can be found in Fig. 6 and Appendix J.1. As shown in Fig. 7, we report the MAE of force estimation under various training data settings. Incorporating position-control generation helps mitigate the challenges from limited angular coverage in the real data and significantly improves performance even with a small subset of real data, especially when the real data covers only a limited range of angles. In Appendix F, we add more experiments and analysis to the force estimations.

E.3. Downstream Task: Pose Estimation

In this section, to demonstrate that CONTROLTAC can generate tactile images aligned with the target contact position, we train a pose estimator using the tactile images generated by CONTROLTAC. To evaluate performance, we train three separate pose estimators: one for a cross, one for three cylinders with varying curvatures and widths, and one for unseen

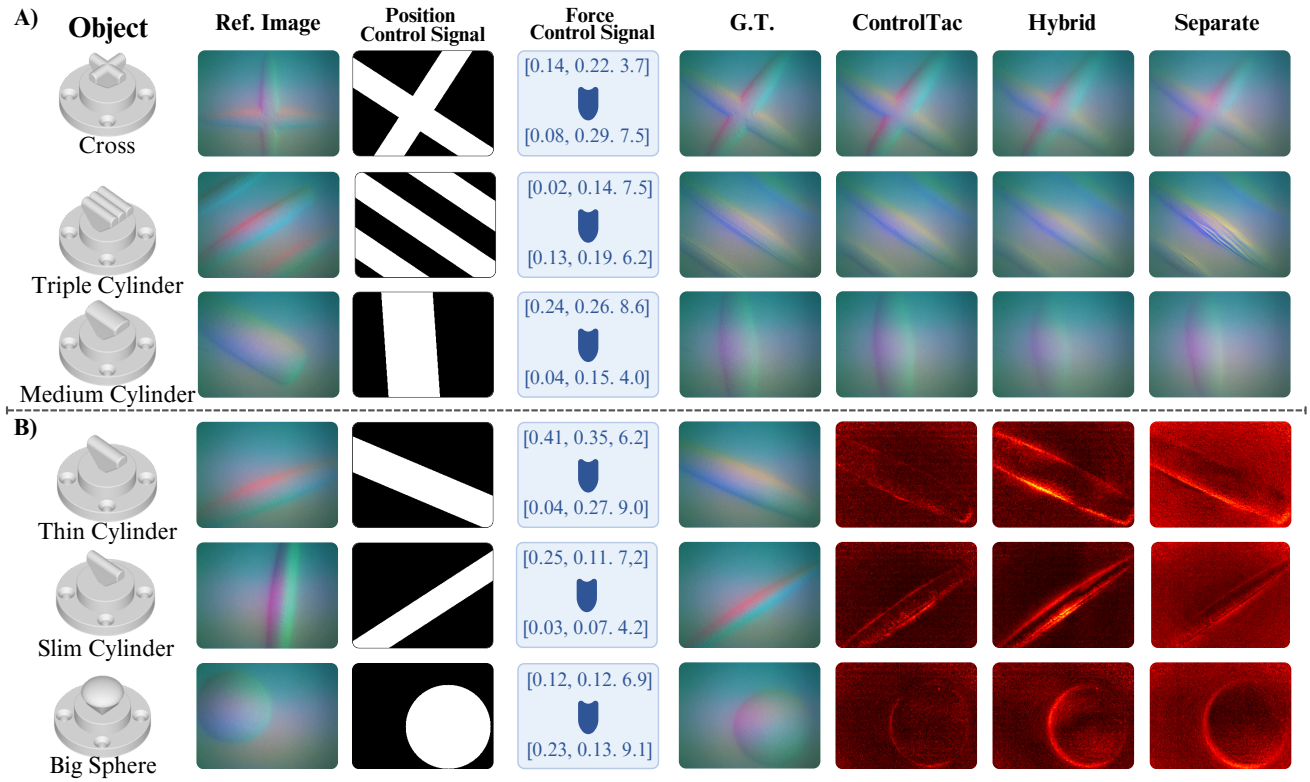

Figure 6. **Qualitative Generation Results.** The first column displays 3D previews of six objects, followed by the input tactile image (Ref. Image) in the second column and the Contact Mask in the third column. The fourth column shows the initial force (top) and target force (bottom). Subsequent columns depict the Ground Truth (G.T.) and results from CONTROLTAC, the hybrid force-position conditional diffusion model (Hybrid), and the separate-control pipeline (Separate). In part `A)`, we visualize the generated images for comparison; in part `B)`, we visualize the error maps highlighting the differences from the ground-truth tactile image. Complete results and force-only generation results are shown in Fig. 11 and Fig. 12 respectively in Appendix J.

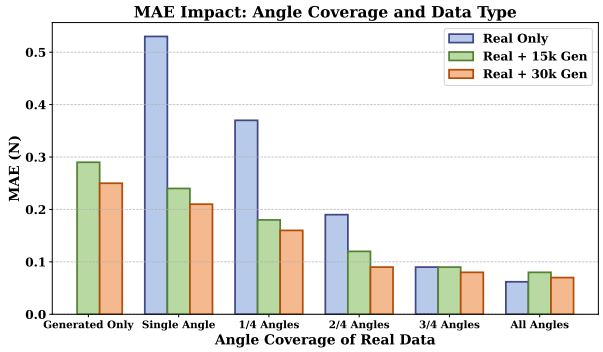

Figure 7. Force estimation performance (MAE) with different data. The sample sizes of 750, 3,750, 7,500, 11,250, and 15,000 are shown as 1, 1/4, 2/4, 3/4, and 4/4 of the contact angles.

shapes, including a T-shape and a type-c USB.

For training the pose estimators, we use a single reference image to generate 30,000 images for each object (5,000 positions and 6 forces). We randomly sample varying number of tactile images from the generated dataset. For the real tactile dataset, each object has 500 unique contact positions, and we randomly select 4-8 forces per position to build a dataset with 3,000 samples for each object. For the unseen T-shape and type-c USB, only generated tactile images are used for evaluation. In the test set, each of the three cylinder types and the cross are annotated with 30 contact positions across multiple force levels from the FeelAnyForce [50] dataset. For the T-shape and type-c USB, we collect 30 contact positions at a single force level. Notably, since we generate the image and label through 2D global mask, the pose label of the image is the centroid of the global mask instead of 2D local area, where it can handle objects which are larger than the sensor.

As shown in Table 5, pose estimators trained solely on tactile images generated by CONTROLTAC achieve strong performance across all objects, including the unseen T Shape and USB. Notably, training with generated data alone often outperforms using simulated data from Taxim [52], which suffers from reduced realism and lower performance, as well as traditional PCA-based methods. Even with a relatively large real dataset, generated data provides significant advantages, since capturing tactile data that fully covers all contact

Table 5. Pose estimation errors (in pixels and degrees) under different settings.

Training Set	X ↓	Y ↓	Angle ↓
Cylinder (3 Types)			
PCA	15	13	22
3,000 real	9	8	4
6,000 real	8	8	4
3,000 sim	21	20	7
12,000 sim	17	15	6
36,000 sim	18	15	6
3,000 gen (fixed)	13	13	6
12,000 gen (fixed)	9	8	5
3,000 gen (unfixed)	9	9	5
6,000 gen (unfixed)	7	6	3
12,000 gen (unfixed)	4	5	3
3,000 real + 3,000 gen	5	4	4
3,000 real + 12,000 gen	3	4	3
3,000 real + 3,000 sim	11	10	5
3,000 real + 12,000 sim	12	13	6
3,000 real + 36,000 sim	14	13	6
Cross			
PCA	56	19	18
1,000 real	7	6	2
3,000 real	6	6	2
1,000 sim	25	23	7
4,000 sim	19	18	5
12,000 sim	18	19	5
1,000 gen (fixed)	11	13	5
4,000 gen (fixed)	7	9	4
1,000 gen (unfixed)	6	9	2
3,000 gen (unfixed)	4	5	2
4,000 gen (unfixed)	3	4	1
1,000 real + 1,000 gen	4	5	1
1,000 real + 4,000 gen	2	4	1
1,000 real + 1,000 sim	15	14	4
1,000 real + 4,000 sim	17	16	4
1,000 real + 12,000 sim	18	14	5
T-shape (Unseen)			
1,000 gen (unfixed)	5	5	4
4,000 gen (unfixed)	4	5	2
USB (Unseen)			
1,000 gen (unfixed)	12	11	4
4,000 gen (unfixed)	8	9	3
16,000 gen (unfixed)	6	6	3
20,000 gen (unfixed)	5	4	3

positions and angles is extremely challenging.

Moreover, combining a small amount of real data with generated data further improves performance. For example, for the Cylinder (3 Types) object, mixing 3,000 real samples with 12,000 generated samples reduces the X and Y errors to 3 and 4 pixels, and the angle error to 3°, outperforming 6,000 real samples or 36,000 simulated samples alone. Similarly, for the Cross object, 1,000 real samples combined with 4,000 generated samples achieve X and Y errors of 2 and 4 pixels, and an angle error of 1°, again surpassing single-source training. In contrast, mixing real data with simulated samples provides limited benefit and, in some cases, can even degrade performance, highlighting the limited realism of simulator data. These results demonstrate that CONTROLTAC-generated data not only complements real data by covering hard-to-capture tactile scenarios, but also maximizes performance gains when used in mixed training, while improving generalization to unseen objects.

We also evaluate the pose estimator under varying versus fixed forces (denoted as "fixed" in Table 5, with the fixed force set to the median value of 6.5 N). Using varying forces leads to better performance, reflecting the natural variation of contact forces during inference. Visual results for Taxim [52] can be found in Figure 13 in Appendix J.4.

E.4. Downstream Task: Object Classification

To evaluate the generalizability of CONTROLTAC and compare it with other data augmentations, we conduct an unseen object classification task. Objects and tactile images are shown in Fig 14 in Appendix J.5.

In this experiment, we select one reference tactile image from each of the six objects and use CONTROLTAC to generate tactile images under varying force and contact positions. For comparison, we consider a traditional data augmentation pipeline [41, 66], which applies geometric and color-based transformations to the selected reference image. The geometric transformations include rotations (eight types at 45° intervals over 360°), flipping (vertical, horizontal, and both), scaling factors (0.8, 1.0, 1.2), and translations along two axes by [-20, 0, 20], yielding 864 augmented images. The color transformations apply hue shifting to synthesis 6 color variants, resulting in 5,184 augmented images.

We evaluate classification performance using three different models: a CNN, a ViT without pre-training, and a ViT with pre-training. We train the models with data samples of size 2,400 and 4,800 using three augmentation methods. The results are summarized in Table 3. Across all models and dataset sizes, CONTROLTAC consistently outperforms traditional augmentation methods, with especially notable improvements in the ViT-based models. This demonstrates the superior utility of conditional tactile generation in enhancing downstream classification performance.

E.5. Case Study: Real-world Experiments

In this section, we utilizes the force estimator and pose estimator by training with augmented dataset in three real-world experiments: Object Pushing, Real-time Pose Tracking, and

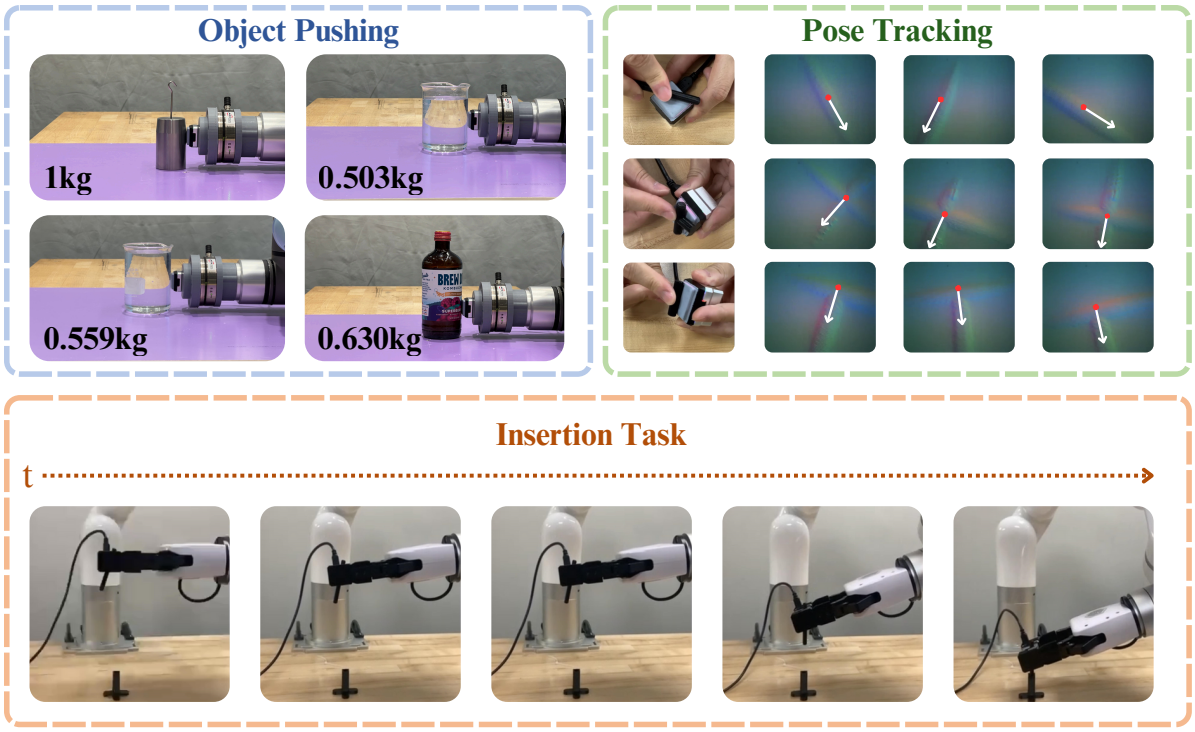

Figure 8. Qualitative examples of real-world experiments.

Precise Insertion.

Object Pushing. In this experiment, we estimate the pushing force between the robot and four objects: a 1 kg calibration weight made of metal, a caliber cylinder full of water (0.559 kg) or almost full (0.503 kg), and a glass bottle of 0.63 kg. We utilize a UR5 robot with a ATI Axia80 force sensor to collect the ground truth forces for pushing. Five pushes of approximately 15s each per object are conducted at a low velocity. The overall experiment setting is shown in Fig. 8.

For evaluation, we compare our model trained on dataset augmented by the force-control generator with the model trained on the real data. As shown in Table 4, the key finding is that the force estimator trained using generated images also reaches similar performance to force estimator trained with the real dataset, which highlights that the force estimator trained with generated data generalizes well to more complex real-world scenarios and new objects with various textures, materials, and weights.

Real-time Pose Tracking. To evaluate the performance of our pose estimator, we conduct a real-time pose tracking experiment. Specifically, we press the printed cylinder, cross, and T-shape object into the sensor and change the object pose by rotating and translating. In this setting, our model can track the pose in real time with 10 Hz, which highlights the practicality of the model trained with our augmented data in this dynamic real-world scenario. A visualization of the task is shown in Fig. 8.

Insertion Task. For the insertion task, we 3D print three different objects (a cylinder, a cross-shaped object, a T-shape object) and a hole. Also, we set up a type-c insertion task for inserting the USB into the charger. We utilize the XArm7 robot arm with two GelSight Mini tactile sensors to accomplish the task using our trained pose estimator. Notably, the tolerance of this insertion task is only 3 mm. See Fig. 8 in Appendix and Appendix H for more details.

To evaluate the performance of our model, we conduct 20 insertion trials for each object type. Our force estimator achieves a success rate of 90% for the cylinder and 85% for both the cross and T-shape objects. These results highlight the practicality of our augmented data, demonstrating that a model trained with data augmented from a single reference image can achieve strong performance on a challenging precise insertion task with *only 3 mm of tolerance*. We further evaluated type-c insertion by training a pose estimator with tactile images generated by ControlTac, achieving an impressive 75% success rate. This demonstrates the effectiveness of our approach. The real object and corresponding tactile images of the Type-C connector are provided in Fig. 15 in Appendix J.6.

F. Additional Experiments

F.1. Robustness Validation of Contact Mask Alignment

To evaluate the sensitivity of position control to inaccuracies in contact mask alignment, we conducted a series of controlled experiments by applying perturbations in three forms: scaling (S), translation (T), and rotation (R). The performance was measured using MSE and SSIM, and the results are summarized in Tables 6 and 7. For individual perturbations, scaling the mask within the range of 0.8 to 1.2 produced only negligible variations in both MSE and SSIM, suggesting that the method is largely insensitive to scale changes. Translation up to 4 pixels and rotation up to 2 degrees, which correspond to typical alignment errors in practice, also resulted in no significant degradation in the quality of the generated outputs. Even when scaling, translation, and rotation perturbations were combined, the generated results remained stable and acceptable. It should be emphasized that the contact mask is primarily used to determine the contact position, whereas the contact area is governed by force control and is therefore unaffected by such perturbations.

Table 6. Individual Perturbation Analysis

Perturbation Type	Parameter	MSE↓	SSIM↑
Scaling (S)	1.0	23	0.83
	1.1	23	0.83
	1.2	24	0.83
	0.9	23	0.83
	0.8	24	0.83
Translation (T)	2 px	23	0.83
	4 px	24	0.83
	6 px	25	0.82
Rotation (R)	2°	23	0.83
	4°	25	0.82
	6°	27	0.82

Table 7. Combined Perturbation Analysis

Perturbation Combination	MSE↓	SSIM↑
S 0.9 + T4 + R2	23	0.82
S 1.1 + T4 + R4	26	0.82
S 0.8 + T6 + R6	28	0.82
S 1.1 + T6 + R6	28	0.82

F.2. Impact of Data Composition on Model Performance

In our experiments, we observe that training the model with a combination of all-angle real data and generated data resulted in slightly worse performance compared to using only real data. This can be explained from two perspectives.

`(1)` First, the performance of the force estimator model is limited by its own capacity. As shown in Table 8, when we trained the model with varying amounts of real data (from 10k to 15k), we found that the MAE improvement plateaued once the data size exceeded 13k, indicating that adding more real data did not lead to significant gains. `(2)` Second, although the generated data is generally of high quality, it inevitably contains small errors. As the proportion of generated data increases, these errors tend to accumulate and negatively impact model training. Specifically, when training solely on different amounts of generated data, the MAE fluctuates as the data size increases, suggesting the presence of error accumulation. Similarly, when mixing real data with a large amount of generated data, model performance is somewhat degraded—for example, the MAE for 15k real + 30k generated is higher than for 15k real + 15k generated.

Nevertheless, the errors in the generated data are minor and do not cause a significant drop in overall model performance. This indicates that while excessive generated data can "dilute" the contribution of real data, it does not fundamentally compromise the results (see Table 8). Furthermore, even when using a much larger amount of generated data (45k or 60k) in combination with real data, the performance does not deteriorate excessively, alleviating concerns about the quality of the generated data.

Table 8. MAE of Force Estimator under Different Data Combinations. `Gen` refers to data generated by CONTROLTAC, while `Real` refers to force estimator trained by real data from FeelAnyForce [50].

Training Data Type	Data Size	MAE ↓
Real	10k	0.061
Real	11k	0.057
Real	12k	0.055
Real	13k	0.051
Real	14k	0.054
Real	15k	0.053
Gen	15k	0.21
Gen	30k	0.17
Gen	45k	0.18
Gen	60k	0.16
Real + Gen	15k + 15k	0.055
Real + Gen	15k + 30k	0.060
Real + Gen	15k + 45k	0.058
Real + Gen	15k + 60k	0.061

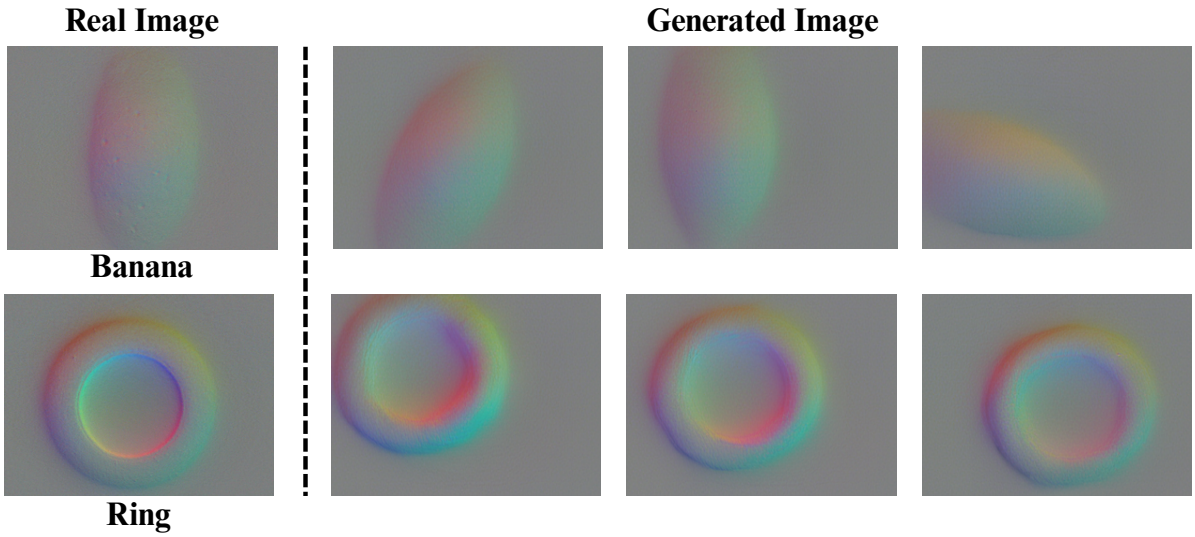

Figure 9. Failure cases on banana and flattened ring.

F.3. Impact of Contact Position Count on ControlTac

In this section, we present the effects of training ControlTac with different numbers of contact positions and using data generated by ControlTac to train the force estimator. The results of our experiments are summarized in Table 9.

Table 9. MAE results based on different contact position counts and data types. `Gen` refers to data generated by ControlTac, while `Real` refers to force estimator trained by real data from FeelAnyForce [50].

Contact Position Count	Data Type	MAE ↓
100	30k Gen	0.272
100	30k Gen + 15k Real	0.077
200	30k Gen	0.207
200	30k Gen + 15k Real	0.067
300	30k Gen	0.174
300	30k Gen + 15k Real	0.060

Table 9 indicates that as the number of contact positions increases, the MAE for models trained solely on generated data decreases. This suggests that more contact positions provide richer feature information, thereby enhancing the model's predictive capability. For instance, with 300 contact positions, the MAE drops to 0.174, a significant improvement over the 0.272 achieved with 100 positions.

Furthermore, incorporating real data results in even lower MAE values, particularly with 100 contact positions, where the MAE decreases from 0.272 to 0.077. This demonstrates the advantage of combining generated data with real data. Similarly, for 200 and 300 contact positions, the inclusion of real data leads to reduced MAE values of 0.067 and 0.060, respectively. This indicates that while generated data can effectively improve model performance, the addition of real data remains essential, especially with smaller datasets.

In summary, increasing the number of contact positions and integrating real data both significantly enhance the performance of the force estimator, suggesting avenues for further optimization in future research.

G. Classifier Architectures

G.1. CNN Classifier

We design a convolutional neural networ (CNN) for image classification, consisting of four convolutional blocks followed by two fully connected layers. The architecture is as follows:

- **Input**: RGB images of shape $(3, 224, 224)$
- **Convolutional Block 1**:
  - Conv2d: $3 \rightarrow 32$, kernel size 3×3, stride 1, padding 1
  - BatchNorm2d
  - ReLU activation
  - MaxPool2d: 2×2
- **Convolutional Block 2**:
  - Conv2d: $32 \rightarrow 64$
  - BatchNorm2d
  - ReLU activation
  - MaxPool2d: 2×2
- **Convolutional Block 3**:
  - Conv2d: $64 \rightarrow 128$
  - BatchNorm2d
  - ReLU activation

  - MaxPool2d: 2×2
- **Convolutional Block 4**:
  - Conv2d: $128 \rightarrow 256$
  - BatchNorm2d
  - ReLU activation
  - MaxPool2d: 2×2
- **Flatten Layer**: Feature map of shape $(256, 14, 14)$ is flattened to (50176)
- **Fully Connected Layers**:
  - Linear: $50176 \rightarrow 512$
  - ReLU + Dropout ($p = 0.5$)
  - Linear: $512 \rightarrow 6$ (number of classes)

G.2. ViT Classifier

We use the Vision Transformer (ViT) architecture [13], specifically the `vit_base_patch16_224` variant implemented via the `timm` library [64]. This transformer-based model operates on image patches and employs self-attention mechanisms.

- **Patch Size**: 16×16
- **Input Resolution**: 224×224
- **Number of Patches**: 196 (i.e., 14×14 patches)
- **Transformer Encoder**:
  - Embedding dimension: 768
  - Number of transformer layers (depth): 12
  - Number of attention heads: 12
  - MLP dimension: 3072
- **Classification Head**: The original head is replaced with:
  - Linear: $768 \rightarrow 6$
- **Pretraining Settings**:
  - *ViT with Pretraining*: The model is initialized with weights pretrained on ImageNet 2012 [10], providing a strong starting point for transfer learning.
  - *ViT without Pretraining*: The model is trained from scratch using random initialization, without access to any external datasets.

H. Details of Precise Insertion

For the precise insertion task, we 3D print three different objects and a hole: a (7 cm-long) cylinder with a diameter of (7 mm), a (7 cm) by (3 cm) cross-shaped object with (7 mm) diameter, a (7 cm) by (3 cm) T shape object with (7 mm) diameter, and a hole measuring (5 cm) in height and (3 cm) in depth with (10 mm) diameter. For the USB insertion task, we let the robot to insert the type-c cable into a charger. To finish the insertion task, we let the XArm7 with two Gelsight Mini grasp the object above the hole with a random angle and in-hand position and then adjust the pose and position according to the estimation result. The setting is shown in Fig. 8.

For the task setting, the hole has been set up in a known position, where the robot can reach the location above it. To finish the insertion, the robot need to adjust it's in-hand pose according to its initial grasping. Specifically, the pose estimator first predict the object's pose on the tactile sensor. Then, we compute the Euclidean distance from the estimated pose to the center. This distance is converted from pixel units to real-world units using a scaling factor of 1 pixel = $\frac{1}{20}mm$. For estimation-based robotic control, the robot adjusts its end-effector by rotating along the Rx axis and translating along the y-axis based on the predicted pose, which align with the object vertically above the hole.

I. Failure Analysis

We acknowledge certain limitations arising from the restricted diversity of the training set, where all contact objects are made of PLA and predominantly exhibit curved surfaces. Consequently, the model shows weaker generation performance for objects with flat surfaces, rich textures, or varying hardness, such as flattened rings and bananas, as illustrated in Fig. 9. For clearer visualization, we subtract the background and apply a constant offset of 127 to shift pixel values into a valid display range. To address these limitations, we augmented the training set of 20,000 samples with 1,000 additional tactile images of flat-surfaced cubes. This targeted addition led to a clear improvement in generation quality for previously unseen flattened rings (MSE reduced from 35 to 27; SSIM increased from 0.80 to 0.83), demonstrating that even a relatively small amount of domain-specific data can substantially enhance performance in underrepresented scenarios.

J. Addition Visualizations

In this section, we provide additional visualizations to clarify the concepts discussed.

J.1. Visualization of Various Angles of Different Objects

In this section, we visualize the various angles of different objects. Fig. 10 provides valuable insights into how angle rotation can lead to dramatic changes in the tactile image's color distribution.

J.2. Visualization of Error Map

In this section, Fig. 11 illustrates the error map of ControlTac compared to two baseline models. It is evident that ControlTac significantly outperforms the other two baseline models.

J.3. Visualization of Generated Image using Force-Control Generation Component

In this section, we showcase the visualization results using the force-control generation component of ControlTac. Fig. 12 presents the generated tactile images for the same contact position, demonstrating excellent results and the effectiveness of this component.

J.4. Analysis of Simulated Tactile Images

In this section, we present tactile images generated using Taxim [52]. As shown in Fig. 13, the simulated images lack realism, highlighting the limitations of current simulation methods for tactile data.

J.5. Object and Tactile Image Visualization for Classification

In this section, we present six objects used in the classification task along with their corresponding tactile images, as shown in Fig. 14.

J.6. Visualization of the Type-C Connector Insertion Task

To better illustrate the Type-C connector insertion task, Fig. 15 shows both the real Type-C connector and its corresponding tactile image used in this experiment.

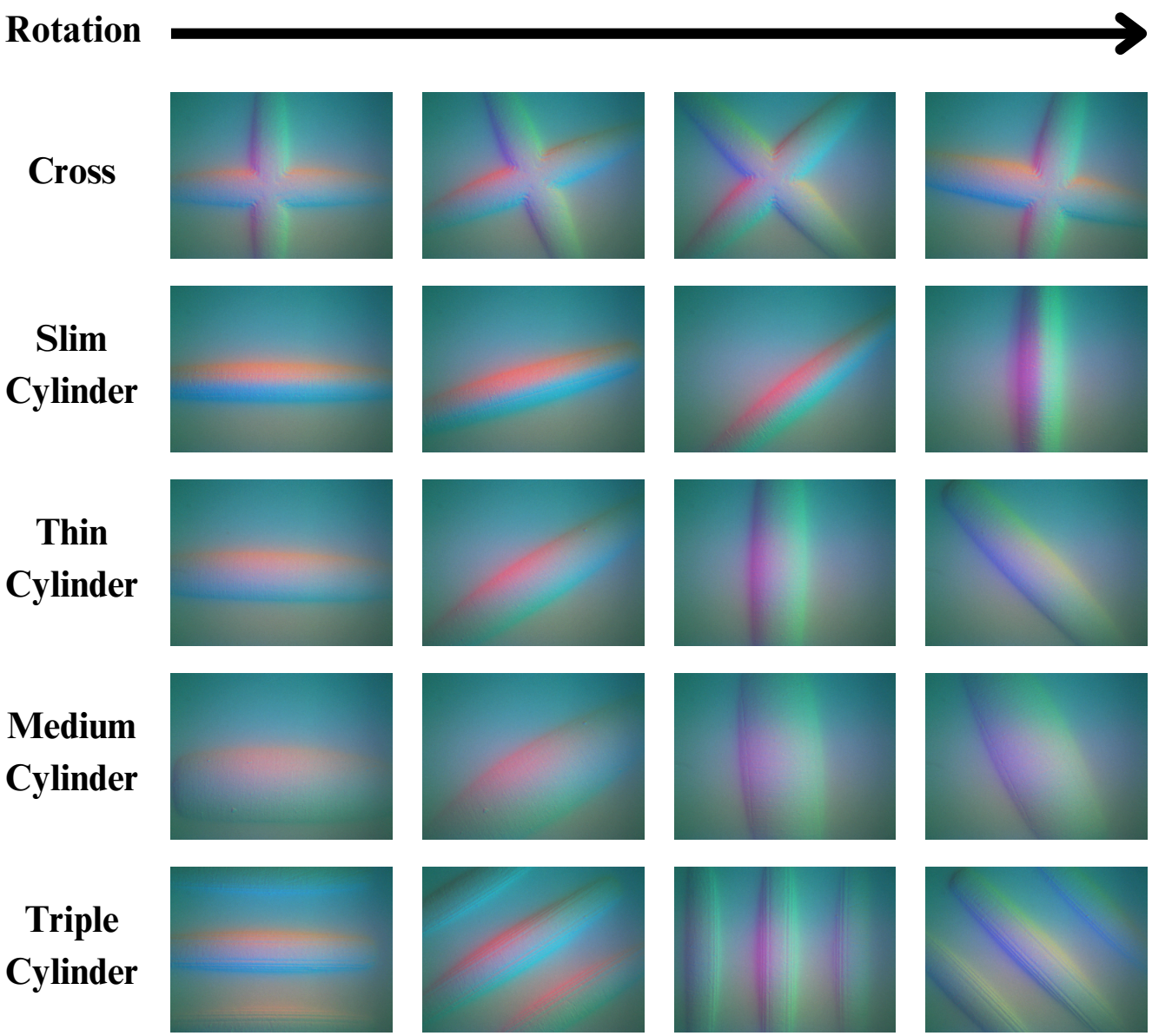

Figure 10. **Visualization of various angles.** Note: The rotational symmetry of spheres renders their angular representations redundant, and thus they are not included here.

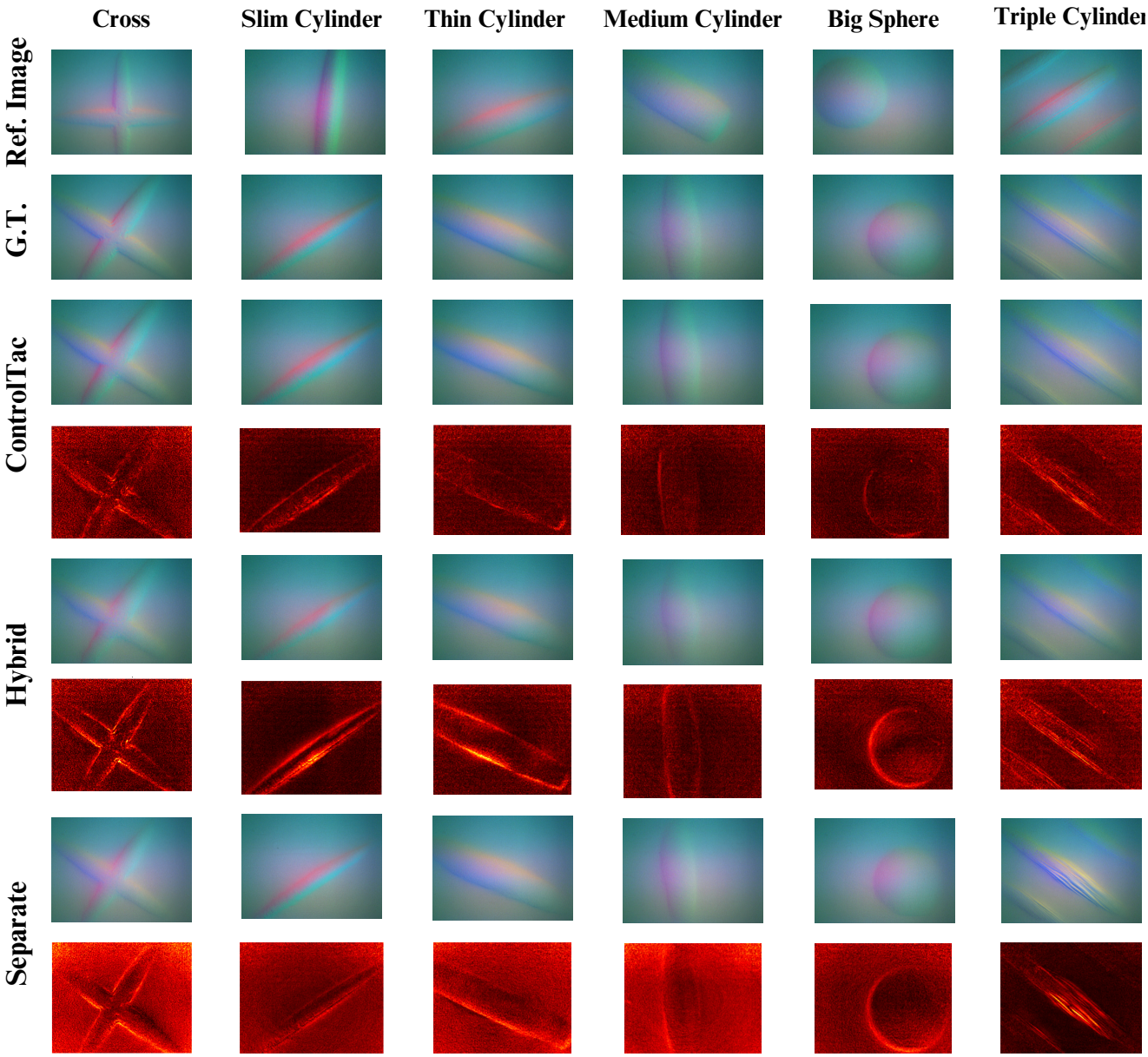

Figure 11. Error map comparison between CONTROLTAC and two baseline models.

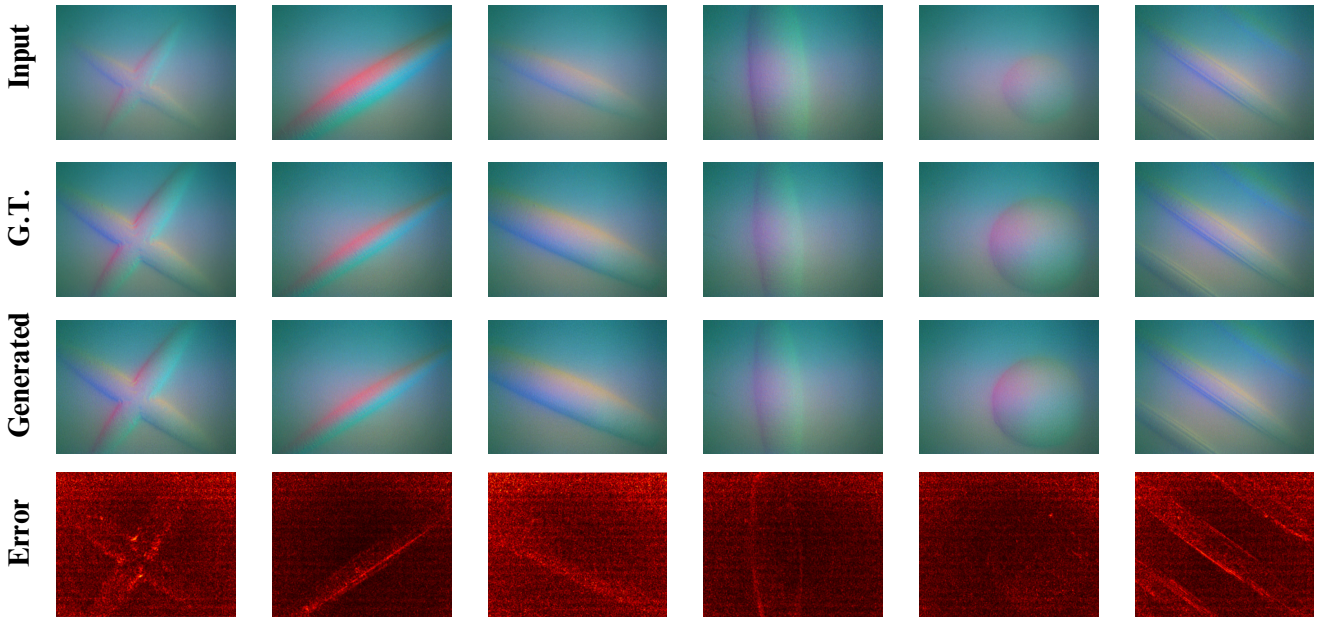

Figure 12. Generated tactile images using the force-control generation component of CONTROLTAC at the same contact position.

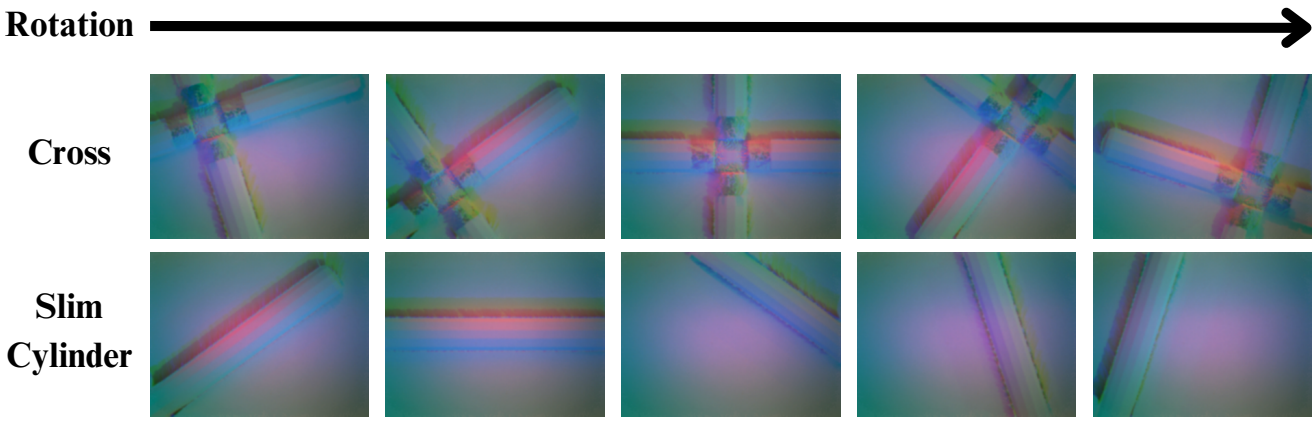

Figure 13. Simulated tactile images using Taxim [52].

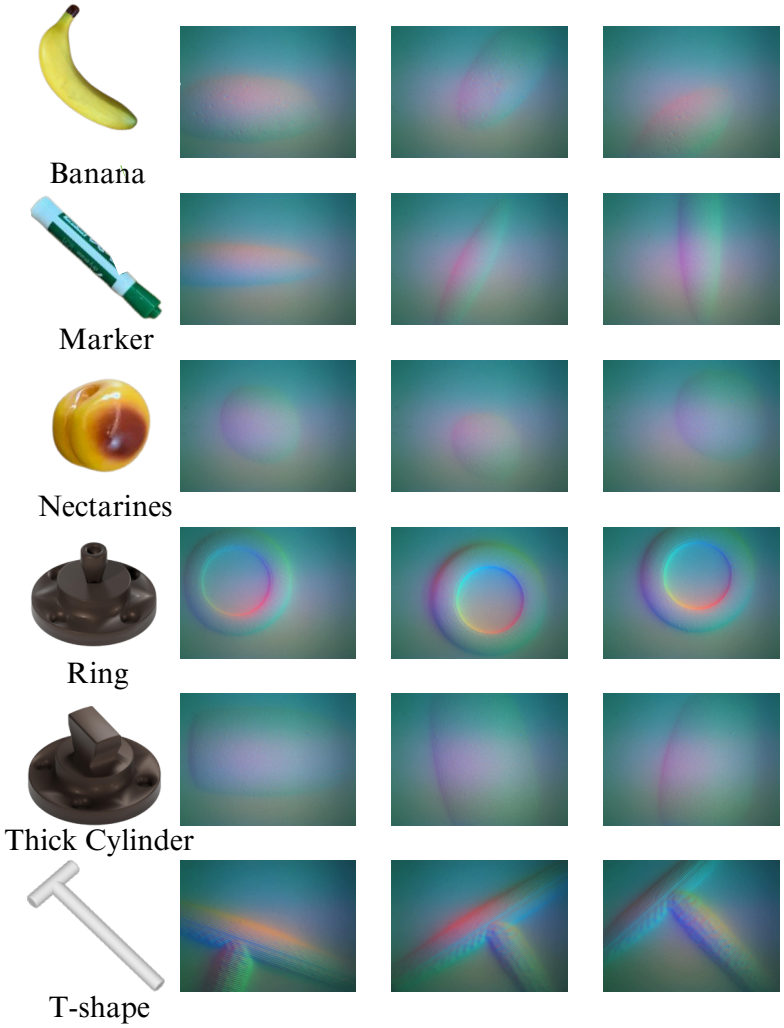

Figure 14. Six objects and their corresponding tactile images used in the classification task.

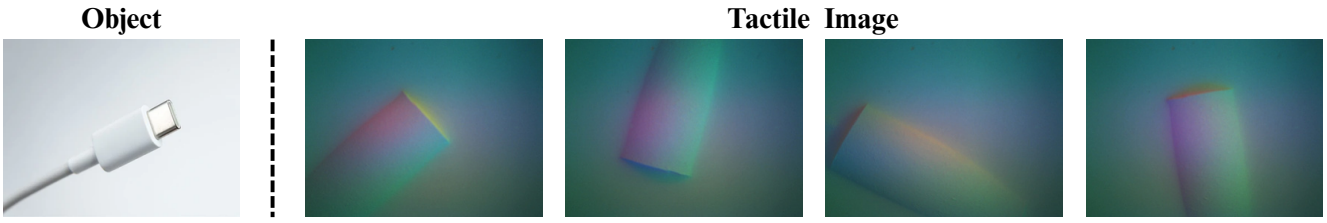

Figure 15. Real Type-C connector and corresponding tactile image.